A METHOD AND APPARATUS FOR ASSOCIATING TICKETS IN A TICKET HIERARCHY

FIELD OF THE INVENTION

[0001] The present invention relates to a method and apparatus for managing tickets and, in particular, to a method and apparatus for associating session tickets allowing associated tickets to be managed as a group.

BACKGROUND OF THE INVENTION

[0002] Conventionally, when users on a client system request access to a resource, a process exists for providing the users with tickets associated with that resource. However, a user may initiate more than one resource request from the same client system and may receive more than one ticket. Management of the user tickets may grow unwieldy and frustrating, especially from a user perspective. For example, a user may focus on a particular session to the exclusion of others and must then re-authenticate itself for the re-issuance of an expired session ticket. A method for simultaneously managing all of the tickets associated with a particular user is desirable.

BRIEF SUMMARY OF THE INVENTION

[0003] The present invention relates to a method and apparatus for associating renewable session tickets. In one aspect, the invention relates to a method and apparatus for associating session tickets and includes a ticketing authority server. The ticketing authority server receives a ticket generation request and information about a client node. It identifies a master session ticket associated in a storage element with the client node. The ticketing authority server then generates a derivative session ticket for the client node and associates the derivative session ticket with the master session ticket. Finally, the ticketing authority server stores information about the client node and the derivative session ticket in the storage element.

[0004] In another aspect, the invention relates to a method and apparatus for renewing associated session tickets. The ticketing authority server receives a session ticket renewal request and a session ticket and retrieves the session associated with the received session ticket. The ticketing authority server then renews the session expiration date for the received session ticket and retrieves the master session ticket associated with the received

session ticket. The ticketing authority server renews the session expiration date of the master session ticket and retrieves any derivative session ticket associated with the master session ticket. Finally, the ticketing authority server renews the session expiration date of the derivative session ticket associated with the master session ticket.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] These and other aspects of this invention will be readily apparent from the detailed description below and the appended drawings, which are meant to illustrate and not to limit the invention, and in which:

[0006] FIG. 1A is a block diagram of an environment suitable for practicing the illustrative embodiment of the present invention;

[0007] FIG. 1B and 1C are block diagrams depicting embodiments of computers useful in connection with the present invention;

[0008] FIG. 1D is a block diagram of an embodiment of a computer network for associating session tickets;

[0009] FIG. 2 is a flow diagram depicting one embodiment of the steps taken by a ticketing authority server to renew each session ticket generated for a client node and associated with a particular master session ticket; and

[0010] FIG. 3 is a flow diagram depicting one embodiment of the steps taken by a ticketing authority server and a policy engine to define and assign session profile types for each session ticket generated for a client node.

DETAILED DESCRIPTION OF THE INVENTION

[0011] The illustrative embodiment of the present invention is applicable to a distributed networking environment where a remote user requests access to content. Prior to discussing the specifics of the present invention, it may be helpful to discuss some of the network environments in which the illustrative embodiment of the present invention may be employed.

[0012] FIG. 1A is a block diagram of an environment suitable for practicing the illustrative embodiment of the present invention. In many embodiments, the ticketing authority server 102, the policy engine 108 and client node 110 are provided as personal computer

or computer servers, of the sort manufactured by the Hewlett-Packard Corporation of Palo Alto, California or the Dell Corporation of Round Rock, TX. Figures 1B and 1C depict block diagrams of a typical computer 100 useful as the client node 110 or the policy engine 108 in those embodiments. As shown in Figures 1B and 1C, each computer 100 includes a central processing unit 102, and a main memory unit 104. Each computer 100 may also include other optional elements, such as one or more input/output devices 130a-130n (generally referred to using reference numeral 130), and a cache memory 140 in communication with the central processing unit 102.

[0013] The central processing unit 102 is any logic circuitry that responds to and processes instructions fetched from the main memory unit 104. In many embodiments, the central processing unit is provided by a microprocessor unit, such as: the 8088, the 80286, the 80386, the 80486, the Pentium, Pentium Pro, the Pentium II, the Celeron, or the Xeon processor, all of which are manufactured by Intel Corporation of Mountain View, California; the 68000, the 68010, the 68020, the 68030, the 68040, the PowerPC 601, the PowerPC604, the PowerPC604e, the MPC603e, the

MPC603ei, the MPC603ev, the MPC603r, the MPC603p, the MPC740, the MPC745, the MPC750, the MPC755, the MPC7400, the MPC7410, the MPC7441, the MPC7445, the MPC7447, the MPC7450, the MPC7451, the MPC7455, the MPC7457 processor, all of which are manufactured by Motorola Corporation of Schaumburg, Illinois; the Crusoe TM5800, the Crusoe TM5600, the Crusoe TM5500, the Crusoe TM5400, the Efficeon TM8600, the Efficeon TM8300, or the Efficeon TM8620 processor, manufactured by Transmeta Corporation of Santa Clara, California; the RS/6000 processor, the RS64, the RS 64 II, the P2SC, the POWER3, the RS64 III, the POWER3-II, the RS 64 IV, the POWER4, the POWER4+, the POWER5, or the POWER6 processor, all of which are manufactured by International Business Machines of White Plains, New York; or the AMD Opteron, the AMD Athalon 64 FX, the AMD Athalon, or the AMD Duron processor, manufactured by Advanced Micro Devices of Sunnyvale, California.

[0014] Main memory unit 104 may be one or more memory chips capable of storing data and allowing any storage location to be directly accessed by the microprocessor 102, such as Static random access memory (SRAM), Burst SRAM or SynchBurst SRAM

(BSRAM), Dynamic random access memory (DRAM), Fast Page Mode DRAM (FPM DRAM), Enhanced DRAM (EDRAM), Extended Data Output RAM (EDO RAM), Extended Data Output DRAM (EDO DRAM), Burst Extended Data Output DRAM (BEDO DRAM), Enhanced DRAM (EDRAM), synchronous DRAM (SDRAM), JEDEC SRAM, PC100 SDRAM, Double Data Rate SDRAM (DDR SDRAM), Enhanced SDRAM (ESDRAM), SyncLink DRAM (SLDRAM), Direct Rambus DRAM (DRDRAM), or Ferroelectric RAM (FRAM).

[0015] In the embodiment shown in FIG. 1B, the processor 102 communicates with main memory 104 via a system bus 120 (described in more detail below). FIG. 1C depicts an embodiment of a computer system 100 in which the processor communicates directly with main memory 104 via a memory port. For example, in FIG. 1C, the main memory 104 may be DRDRAM.

[0016] FIG. 1B and FIG. 1C depict embodiments in which the main processor 102 communicates directly with cache memory 140 via a secondary bus, sometimes referred to as a "backside" bus. In other embodiments, the main processor 102 communicates with cache memory 140 using the system bus 120. Cache memory 140

typically has a faster response time than main memory 104 and is typically provided by SRAM, BSRAM, or EDRAM.

[0017] In the embodiment shown in FIG. 1B, the processor 102 communicates with various I/O devices 130 via a local system bus 120. Various busses may be used to connect the central processing unit 102 to the I/O devices 130, including a VESA VL bus, an ISA bus, an EISA bus, a MicroChannel Architecture (MCA) bus, a PCI bus, a PCI-X bus, a PCI-Express bus, or a NuBus. For embodiments in which the I/O device is a video display, the processor 102 may use an Advanced Graphics Port (AGP) to communicate with the display. FIG. 1C depicts an embodiment of a computer system 100 in which the main processor 102 communicates directly with I/O device 130b via HyperTransport, Rapid I/O, or InfiniBand. FIG. 1C also depicts an embodiment in which local busses and direct communication are mixed: the processor 102 communicates with I/O device 130a using a local interconnect bus while communicating with I/O device 130b directly.

[0018] A wide variety of I/O devices 130 may be present in the computer system 100. Input devices include keyboards, mice,

trackpads, trackballs, microphones, and drawing tablets. Output devices include video displays, speakers, inkjet printers, laser printers, and dye-sublimation printers. An I/O device may also provide mass storage for the computer system 100 such as a hard disk drive, a floppy disk drive for receiving floppy disks such as 3.5-inch, 5.25-inch disks or ZIP disks, a CD-ROM drive, a CD-R/RW drive, a DVD-ROM drive, tape drives of various formats, and USB storage devices such as the USB Flash Drive line of devices manufactured by Twintech Industry, Inc. of Los Alamitos, California.

[0019] In further embodiments, an I/O device 130 may be a bridge between the system bus 120 and an external communication bus, such as a USB bus, an Apple Desktop Bus, an RS-232 serial connection, a SCSI bus, a FireWire bus, a FireWire 800 bus, an Ethernet bus, an AppleTalk bus, a Gigabit Ethernet bus, an Asynchronous Transfer Mode bus, a HIPPI bus, a Super HIPPI bus, a SerialPlus bus, a SCI/LAMP bus, a FibreChannel bus, or a Serial Attached small computer system interface bus.

[0020] General-purpose desktop computers of the sort depicted in FIG. 1B and FIG. 1C typically operate under the control of operating systems, which control scheduling of tasks and access

to system resources. Typical operating systems include: MICROSOFT WINDOWS, manufactured by Microsoft Corp. of Redmond, Washington; MacOS, manufactured by Apple Computer of Cupertino, California; OS/2, manufactured by International Business Machines of Armonk, New York; and Linux, a freely-available operating system distributed by Caldera Corp. of Salt Lake City, Utah, among others.

[0021] The client node 110 may be any personal computer (e.g., 286, 386, 486, Pentium, Pentium II, Macintosh computer), Windows-based terminal, Network Computer, wireless device, information appliance, RISC Power PC, X-device, workstation, mini computer, main frame computer, personal digital assistant, or other computing device that has a windows-based desktop and sufficient persistent storage for executing a small, display presentation program. The display presentation program uses commands and data sent to it across communication channels to render a graphical display. Windows-oriented platforms supported by the client node 110 can include, without limitation, WINDOWS 3.x, WINDOWS 95, WINDOWS 98, WINDOWS NT 3.51, WINDOWS NT 4.0, WINDOWS 2000, WINDOWS CE, MAC/OS, Java, and UNIX. The client node 110

can include a visual display device (e.g., a computer monitor), a data entry device (e.g., a keyboard), persistent or volatile storage (e.g., computer memory) for storing downloaded application programs, a processor, and a mouse. Execution of a small, display presentation program allows the client node 110 to participate in a distributed computer system model (i.e., a server-based computing model).

[0022] For embodiments in which the client node 110 is a mobile device, the device may be a JAVA-enabled cellular telephone, such as the i50sx, i55sr, i58sr, i85s, i88s, i90c, i95cl, or the im11000, all of which are manufactured by Motorola Corp. of Schaumburg, Illinois, the 6035 or the 7135, manufactured by Kyocera of Kyoto, Japan, or the i300 or i330, manufactured by Samsung Electronics Co., Ltd., of Seoul, Korea. In other embodiments in which the client node 110 is mobile, it may be a personal digital assistant (PDA) operating under control of the PalmOS operating system, such as the Tungsten W, the VII, the VIIx, the i705, all of which are manufactured by palmOne, Inc. of Milpitas, California. In further embodiments, the client node 110 may be a personal digital assistant (PDA) operating under control of the PocketPC operating system, such as the iPAQ 4155, iPAQ 5555,

iPAQ 1945, iPAQ 2215, and iPAQ 4255, all of which manufactured by Hewlett-Packard Corporation of Palo Alto, California, the ViewSonic V36, manufactured by ViewSonic of Walnut, California, or the Toshiba PocketPC e405, manufactured by Toshiba America, Inc. of New York, New York. In still other embodiments, the client node is a combination PDA/telephone device such as the Treo 180, Treo 270 or Treo 600, all of which are manufactured by palmOne, Inc. of Milpitas, California. In still further embodiments, the client node 102 is a cellular telephone that operates under control of the PocketPC operating system, such as the MPx200, manufactured by Motorola Corp. A user of the client node 110 may communicate with the other network elements using protocols such as the depicted Hypertext Transport Protocol Secure (HTTPS) request 115, or an HTTP (Hypertext Transport Protocol) or FTP (File Transport Protocol) request.

[0023] FIG. 1D depicts one embodiment of a ticket system 100 constructed in accordance with the invention is depicted, which includes a ticketing authority server 102, a ticket storage element 104, a policy engine 106, and a session ticket profile type 108. Although only one ticketing authority server 102, ticket storage

element 104, and policy engine 106, are depicted in the embodiment shown in Figure 1D, it should be understood that the system may provide multiple ones of any or each of those components. In some of these embodiments, the servers may be geographically dispersed. In some embodiments, the ticketing authority server 102 may further comprise the ticket storage element 104. In other embodiments, the policy engine 106 may further comprise the ticketing authority server 102. In still other embodiments, the policy engine 106 may comprise the ticketing authority server 102 and the ticket storage element 104.

[0024] In brief overview, the ticketing authority server 102 associates each session ticket generated for a client node with a master session ticket to manage session tickets and the entries in the ticket storage element 104 to which they refer. In some embodiments, the ticketing authority server 102 may comprise application software executing on a general-purpose computer, such as the ones described above in connection with FIG. 1B and FIG. 1C. In other embodiments, the ticketing authority server 102 may comprise special purpose hardware. In still other

embodiments, the ticketing authority server 102 may be a process executing on a policy engine 108.

[0025] In more detail, the ticketing authority server 102 receives a ticket generation request and information about a client node. In some embodiments, the ticketing authority server 102 receives the ticket generation request from a proxy server. In other embodiments, the ticketing authority server 102 receives the ticket generation request from an authentication server. In those embodiments, the authentication server may request a specific type of session ticket based upon an access control decision made by the authentication server.

[0026] In some embodiments, the ticketing authority server 102 receives the ticket generation request over a network connection. The network can be a local area network (LAN), a metropolitan area network (MAN), or a wide area network (WAN) such as the Internet. The client node 102 and the policy engine 106 may connect to a network through a variety of connections including standard telephone lines, LAN or WAN links (e.g., T1, T3, 56 kb, X.25), broadband connections (ISDN, Frame Relay, ATM), and wireless connections. Connections between the client node 102 and

the policy engine 106 may use a variety of data-link layer communication protocols (e.g., TCP/IP, IPX, SPX, NetBIOS, NetBEUI, SMB, Ethernet, ARCNET, Fiber Distributed Data Interface (FDDI), RS232, IEEE 802.11, IEEE 802.11a, IEE 802.11b, IEEE 802.11g and direct asynchronous connections).

[0027] In one embodiment, a session ticket is a reference to an entry in a storage element 104. In one embodiment, the session ticket comprises a random number. In other embodiments, the session ticket is a hash of client node information.

[0028] In one embodiment, the entry in the storage element 104 may include, without limitation, an identifier for a master session ticket, an identifier for a derivative session ticket, a session profile type, and client node information. In one embodiment, the storage element 104 is a database. In another embodiment, the storage element is a data structure. In yet another embodiment, the storage element 104 is a file. In other embodiments, the storage element 104 is a relational database.

[0029] In one embodiment, the ticketing authority server 102 receives information about the client node requiring the session ticket. The client node information may include, without

limitation, the user name of the user accessing the client node, the network domain on which the client node resides, and the type of session initiated.

[0030] The ticketing authority server 102 identifies a master session ticket associated in the ticket storage element 104 with the client node and generates a derivative session ticket for the client node. In one embodiment, the ticketing authority server 102 identifies the master session ticket by using the client information in the ticket storage element 104 query to determine whether any master session tickets exist. In one embodiment, the ticket storage element 104 stores an entry for each master session tickets including, without limitation the client node identifier, session profile type, an identifier for the master session ticket and the identifier of each derivative session ticket associated with the master session ticket. In this embodiment, the ticketing authority server 102 associates the derivative session ticket with the master session ticket by adding an identifier for the derivative session ticket in the ticket storage element 104 entry for the master session ticket.

[0031] In another embodiment, the ticketing authority server 102 determines that the client node has not received a master session ticket. In this embodiment, the ticketing authority server 102 generates a master session ticket for the client. The ticketing authority server 102 issues the session ticket and stores in the ticket storage element 104 the received client node information and an identifier for the master session ticket.

[0032] Finally, the ticketing authority server 102 stores information about the client node and the derivative session ticket in the ticket storage element 104. In one embodiment, the ticketing authority server 102 writes an entry to the ticket storage element 104 storing the received client node information, a session profile type, an identifier for the derivative session ticket, and an identifier for the associated master session ticket in the ticket storage element 104.

[0033] Referring now to FIG. 2, a flow diagram depicts one embodiment of the steps taken by the ticketing authority server 102 to renew associated session tickets. The ticketing authority server 102 receives a request for session ticket renewal and a session ticket (Step 202) and renews the expiration date for that session

ticket and all session tickets associated with that received session ticket.

[0034] The ticketing authority server 102 retrieves the session associated with the received session ticket (Step 204). In one embodiment, the ticketing authority server 102 queries the ticket storage element 104 to retrieve the relevant session information. In this embodiment, the session ticket includes the identifiers needed to retrieve the session information.

[0035] The ticket server 102 renews the expiration date for the received session ticket (Step 206). In some embodiments, the expiration date is part of a session profile type. In other embodiments, the expiration date may be an entry in the ticket storage element 104.

[0036] The ticketing authority server 102 then renews the expiration dates for all session tickets associated with the received session ticket (Step 208). In one embodiment, the ticketing authority server 102 retrieves the master session ticket associated with the received session ticket. In some embodiments, the ticketing authority server 102 retrieves the master session ticket

from the ticket storage element 104 using the session ticket received with the renewal request.

[0037] In one embodiment, the ticketing authority server 102 then retrieves the derivative session tickets associated with the received session ticket. In some embodiments, the ticketing authority server 102 retrieves the derivative session tickets from the ticket storage element 104 by first retrieving the master session ticket and then querying the ticket storage element 104 for derivative session tickets associated with the master session ticket.

[0038] In one embodiment, a network element providing a resource to the client node may request the master session ticket from the client node. In this embodiment, the network element periodically requests from the ticketing authority server 102 the expiration date associated with the master session ticket. Additionally, in this embodiment, the network element monitors the client node level of activity periodically. Before the master session ticket expires, if the client node has not idled, the network element requests, on behalf of the client node, that the ticketing authority server 102 renew the master session ticket and the related derivative session ticket or tickets.

[0039] In an example of this embodiment, the client node receives a renewable master session ticket and receives two renewable derivative session tickets. In some embodiments, one derivative session is, for example, a connection using a presentation-layer protocol such as the Independent Computing Architecture (ICA) protocol, available from Citrix Systems, Inc. of Fort Lauderdale, Florida, or the Remote Desktop Protocol (RDP), manufactured by Microsoft Corporation. In other embodiments, the derivative session is a connection using a presentation-layer protocol wrapped in a higher protocol.

[0040] The client node remains active in one such derivative session but becomes idle in the other derivative session ticket and in the master session ticket. The network element managing the ICA session determines that the idle derivative session has an imminent expiration date. The network element requests session ticket renewal for the derivative and master session tickets since the client remains active in the ICA session. The ticketing authority server 102 verifies the validity of each ticket and that each session ticket profile type allows renewal of the session ticket. The

ticketing authority server 102 then renews all the session tickets for that client node.

[0041] Referring now to FIG. 3, a flow diagram depicts one embodiment of the steps taken by the ticketing authority server 102 and the policy engine 106 to define and assign session profile types 108 for session tickets. In brief overview, the policy engine 106 configures at least one session profile type 108 and the ticketing authority server 102 assigns a session profile type 108 to each session ticket it generates.

[0042] In more detail, the policy engine 106 configures at least one session profile type 108. In one embodiment, the session profile type comprises configurable parameters. In one embodiment, one configurable parameter defines session lifetimes. In this embodiment, the session profile type 108 defines an expiration date for the session ticket and the session expires when the session ticket expires.

[0043] In other embodiments, one configurable parameter defines session failure reconnection rights. In one of these embodiments, a session failure reconnection right allows the

session ticket to be a multiuse session ticket. In this embodiment, an idle timer controls the session lifetime.

[0044] In another of these embodiments, a session failure reconnection right allows the session ticket to be renewed. In this embodiment, the ticketing authority server 102 issues both a derivative session ticket and a reconnect session ticket. When the derivative session ticket expires, the client node may use the reconnect session to renew the derivative session ticket. In this embodiment, a separate idle timer monitors the length of time that passes between expiration and the presentation of the reconnect session ticket. If an unacceptable amount of time passes, the ticketing authority server 102 rejects the reconnect session ticket.

[0045] In another of these embodiments, a session failure reconnection right allows the session ticket to be used only once. In this embodiment, the ticketing authority server 102 will not renew this single use session ticket. Upon expiration of the session ticket, the session also expires and may not be renewed.

[0046] In another embodiment, the session profile type 108 defines the authorization credentials required from the client node for a session ticket to issue. In some embodiments, the

authorization credentials include a number of types of authentication information, including without limitation, user names, client names, client addresses, passwords, PINs, voice samples, one-time passcodes, biometric data, digital certificates, etc. and combinations thereof.

[0047] In one embodiment, the policy engine 106 defines which of the rights are available for a particular session profile type 108 (Step 302). When the ticketing authority server 102 generates a session ticket, it uses the client node information to determine the session profile type 108 to assign to the session ticket (Steps 304 and 306). In one embodiment, the session profile type 108 grants differing session access rights for each session ticket the client node obtains, but the client node need only authenticate itself one time. This embodiment allows the client node to avoid multiple authentication procedures while enabling differing levels of access for various session types. Finally, the ticketing authority server 102 stores client node information in the ticket storage element 104 (Step 308) and transmits the issued session ticket to the originator of the ticket generation request (Step 310).

[0048] The present invention may be provided as one or more computer-readable programs embodied on or in one or more articles of manufacture. The article of manufacture may be a floppy disk, a hard disk, a compact disc, a digital versatile disc, a flash memory card, a PROM, a RAM, a ROM, or a magnetic tape. In general, the computer-readable programs may be implemented in any programming language. Some examples of languages that can be used include C, C++, C#, or JAVA. The software programs may be stored on or in one or more articles of manufacture as object code.

[0049] While the invention has been shown and described with reference to specific preferred embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the following claims.